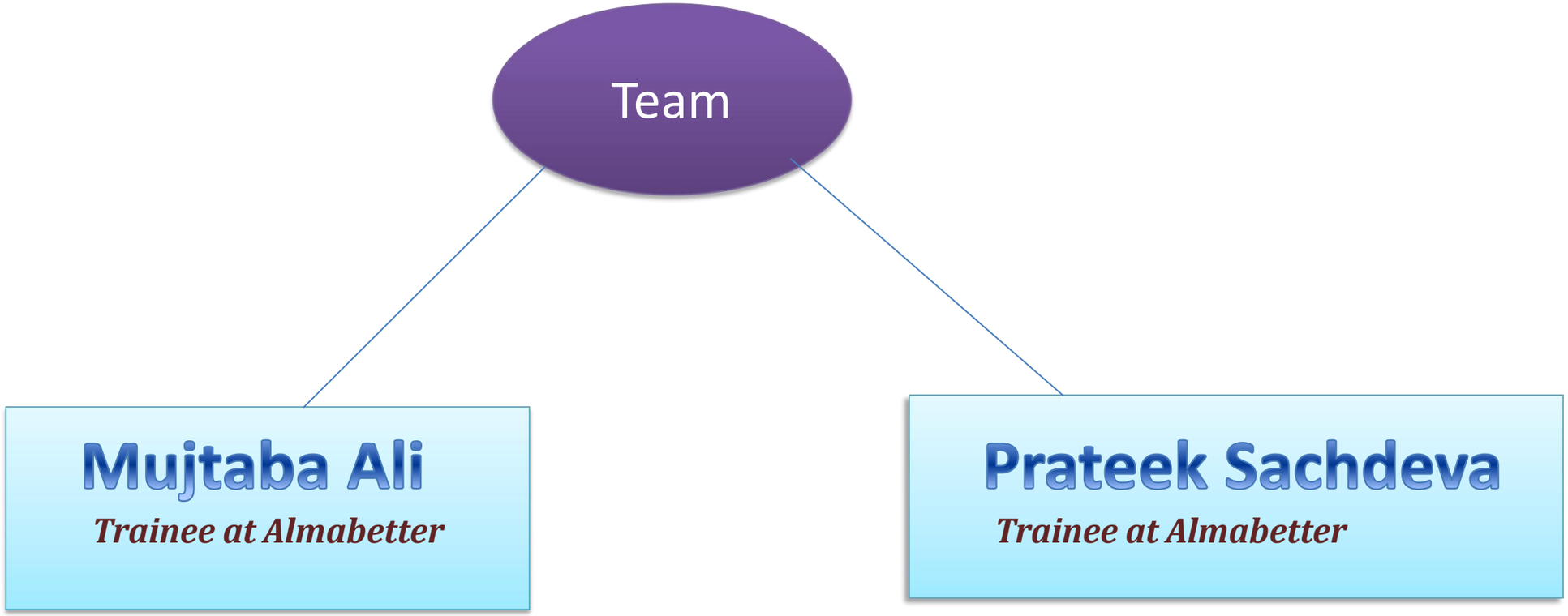
Capstone Project - Classification

Cardiovascular Risk Prediction


Team
Mujtaba Ali
Trainee at Almabetter
Prateek Sachdeva
Trainee at Almabetter

❑ Introduction

- *The dataset is from an ongoing cadiovascular study on residents of the town of Framingham, Masschusetts. The classification goal is to predict whether the patient has a 10-year risk of future coronary heart disease (CHD). The dataset provides the patients information. It includes over 4000 records and 15 attributes. Each attribute is a potential risk factor. There are both demographic, behavioral, and medical risk factors.*

- *The goal of the Cardiovascular Health Study (CHS) is to identify risk factors for cardiovascular disease related to the onset of coronary heart disease and stroke in adults.*

- *As with men, women's most common heart attack symptom is chest pain (angina) or discomfort. But women may experience other symptoms that are typically less associated with heart attack, such as shortness of breath, nausea/vomiting, and back or jaw pain. Learn about the warning signs of heart attack in women.*

❑ Data Summary

- *In this session, we will have the overview of the basic understanding of our dataset variables. What does particular features means and how it's distributed, what type of data is it. There is a dataset in Cardiovascular project, and Cardiovascular dataset is having 17 columns in total. We can get this by basic inspection of our dataset.*

- *Initially the total records are 3390.*

	id	age	education	sex	is_smoking	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
0	0	64	2.00	F	YES	3.00	0.00	0	0	0	221.00	148.00	85.00	NaN	90.00	80.00	1
1	1	36	4.00	M	NO	0.00	0.00	0	1	0	212.00	168.00	98.00	29.77	72.00	75.00	0
2	2	46	1.00	F	YES	10.00	0.00	0	0	0	250.00	116.00	71.00	20.35	88.00	94.00	0
3	3	50	1.00	M	YES	20.00	0.00	0	1	0	233.00	158.00	88.00	28.26	68.00	94.00	1
4	4	64	1.00	F	YES	30.00	0.00	0	0	0	241.00	136.50	85.00	26.42	70.00	77.00	0

❑ Understand the variables

1. ***Sex:***
   - Male(0) or female(1);(Nominal).

2. Age:
   - Age of the patient;(Continuous - Although the recorded ages have been truncated to whole numbers, the concept of age is continuous) education.

3. CurrentSmoker *:*
   - Whether or not the patient is a current smoker (Nominal).

4.CigsPerDay :
   - The number of cigarettes that the person smoked on average in one day.(can be considered continuous as one can have any number of cigarretts, even half a cigarette.).

5.BPMeds *:*
   - Whether or not the patient was on blood pressure medication (Nominal).

6. PrevalentStroke:

- Whether or not the patient had previously had a stroke (Nominal).

7. **PrevalentHyp:**

- Whether or not the patient was hypertensive (Nominal).

8. **Diabetes:**

- Whether or not the patient had diabetes (Nominal).

9. **TotChol :**

- Total cholesterol level (Continuous).

10. SysBP:

- Systolic blood pressure (Continuous).

11**. DiaBP:**

- Diastolic blood pressure (Continuous).

12. **BMI:**

- Body Mass Index (Continuous).

13. HeartRate:

- Heart rate (Continuous - In medical research, variables such as heart rate though in fact discrete, yet are considered continuous because of large number of possible values.).

14. Glucose :

- Glucose level (Continuous).

15. 10 year risk of coronary heart disease CHD :

- 10 year risk of coronary heart disease CHD (binary: “1”, means “Yes”, “0” means “No”) - Target Variable.

❑ Project Architecture:

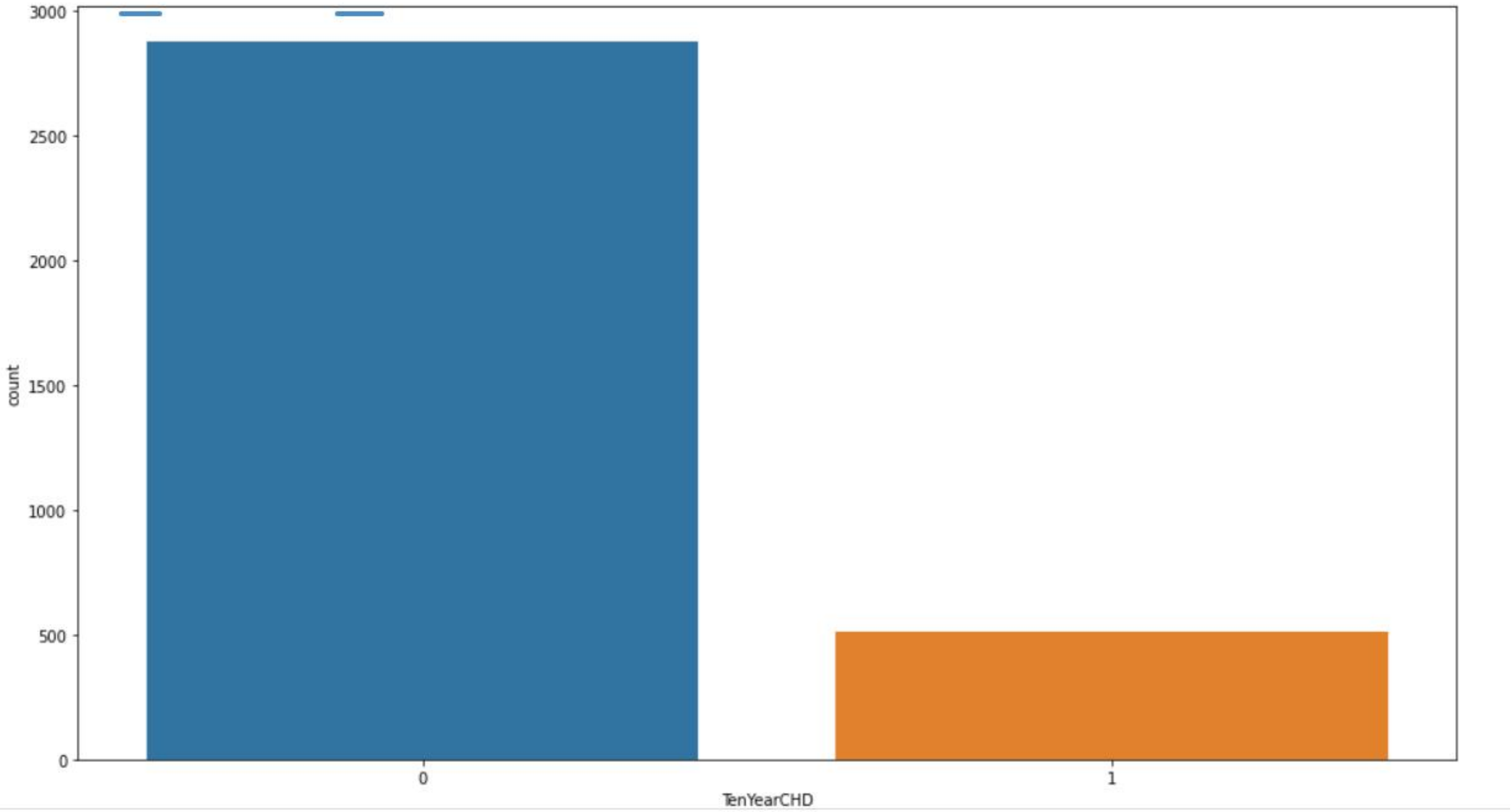
1. EDA:

✓ ***Understanding business problem:*** *This section, we try to understand the problem.*

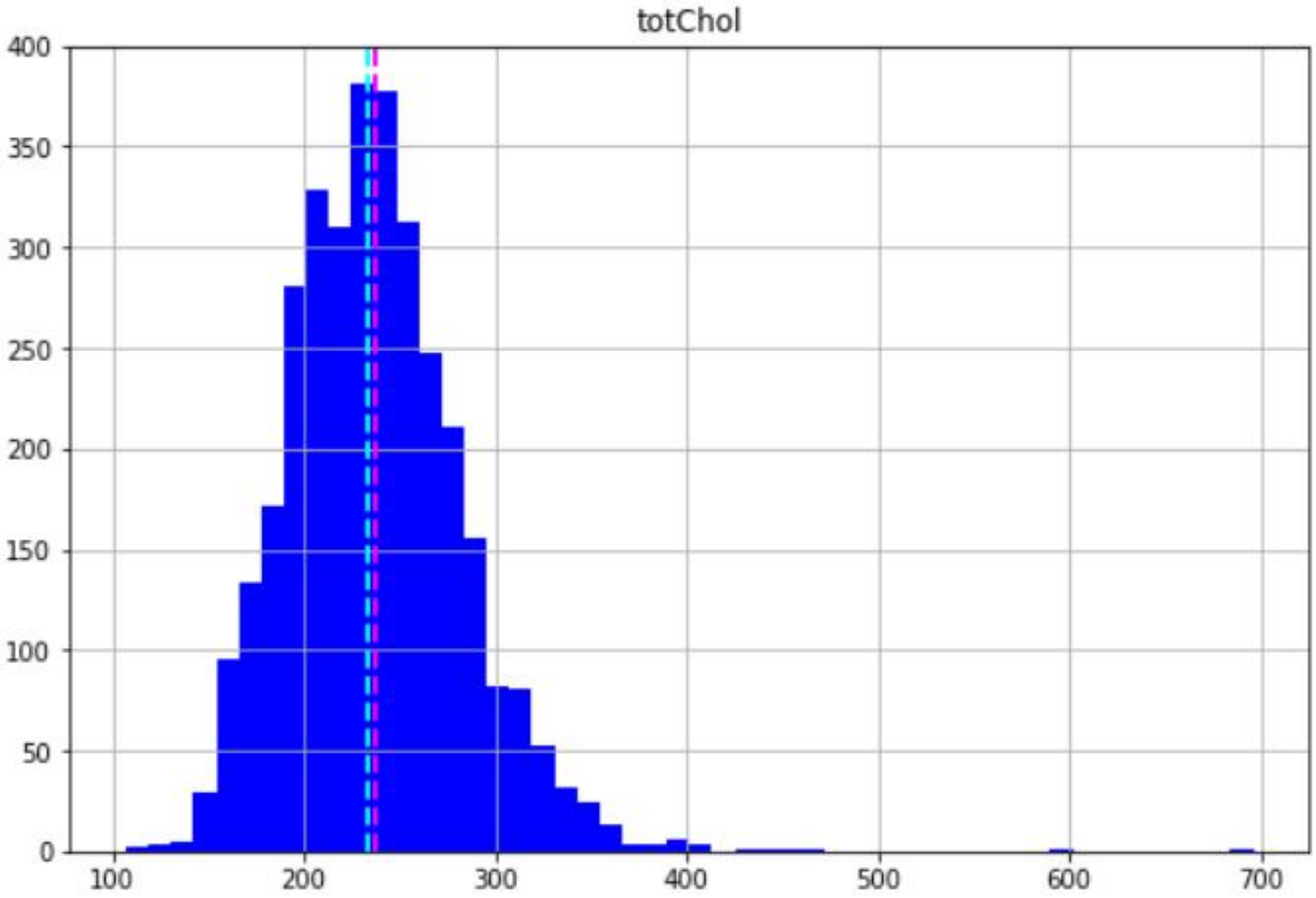
✓ ***Visualization and Analyzing relationships:*** *In this section, we try to understand the distribution between dependent and independent features. We plotted the relationships using various plots like bar plot, histogram, lineplot, and scatterplot etc. We found some insights and relationships like Cigaretteperday, BPMeds, prevalentStroke, Diabetes, totChol, and glucose have right skewed distribution.*
SysBP, DiaBP, BMI, and heartRate have lightly right skewed distribution.

- *We can see from cigsPerDay value counts that there are 1703 people who have consumed 0 cigarette per day. The same insight, we can extract from is_smoking_YES column. So we can easily drop is_smoking_YES column since both of them are giving same information*

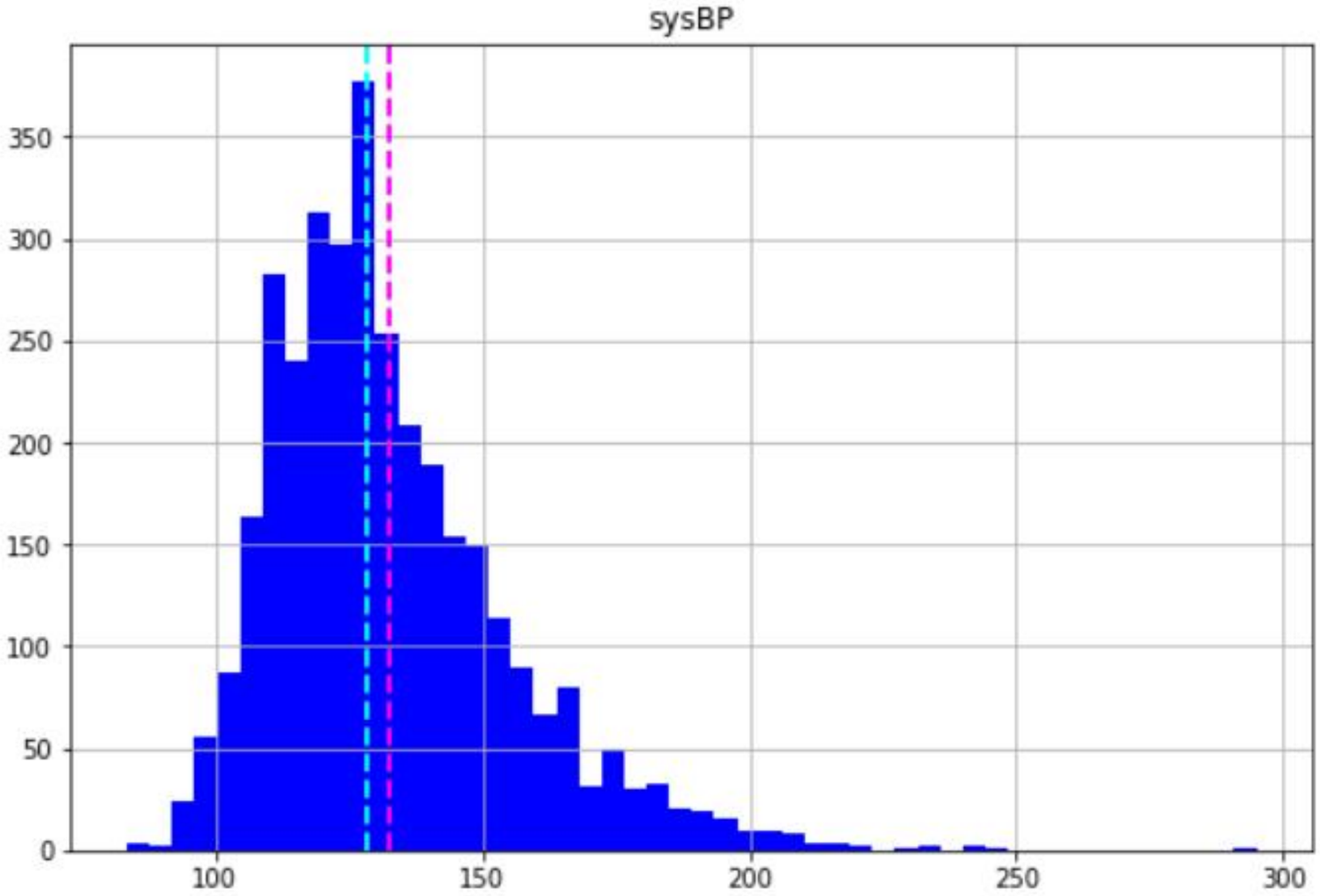
The count plot of target variable which is imbalanced.



Showing the distribution of Total Cholesterol using histogram of seaborn. And we can see that it is almost normally distributed.



Visualizing the distribution of systolic blood pressure using histplot of seaborn. And we can see that it is right skewed data.



2. Clean-Up:

- ***Missing Values:*** *We found the missing values in some columns. We imputed the data using KNNImputer.*

Finding the missing value percentage in each column:

	Missing_Values	Percentage
glucose	304	9.00
education	87	3.00
BPMeds	44	1.00
totChol	38	1.00
cigsPerDay	22	1.00
BMI	14	0.00
heartRate	1	0.00
age	0	0.00

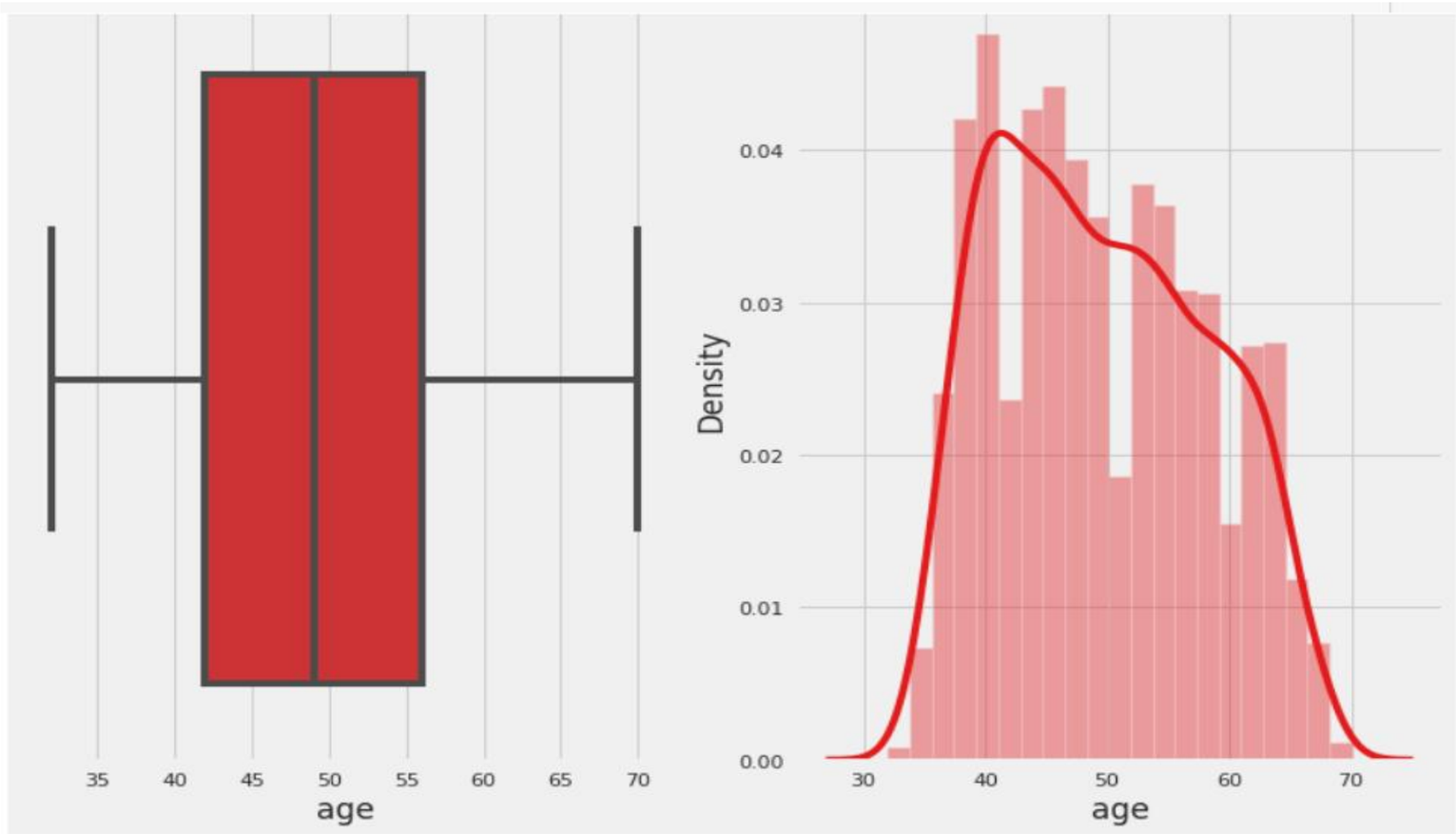
After imputing the missing values, we can see that there is no null value left in any column.

	Missing_Values	Percentage	Dtype
age	0	0.0	float64
education	0	0.0	float64
cigsPerDay	0	0.0	float64
BPMeds	0	0.0	float64
prevalentStroke	0	0.0	float64
prevalentHyp	0	0.0	float64
diabetes	0	0.0	float64
totChol	0	0.0	float64
BMI	0	0.0	float64
heartRate	0	0.0	float64

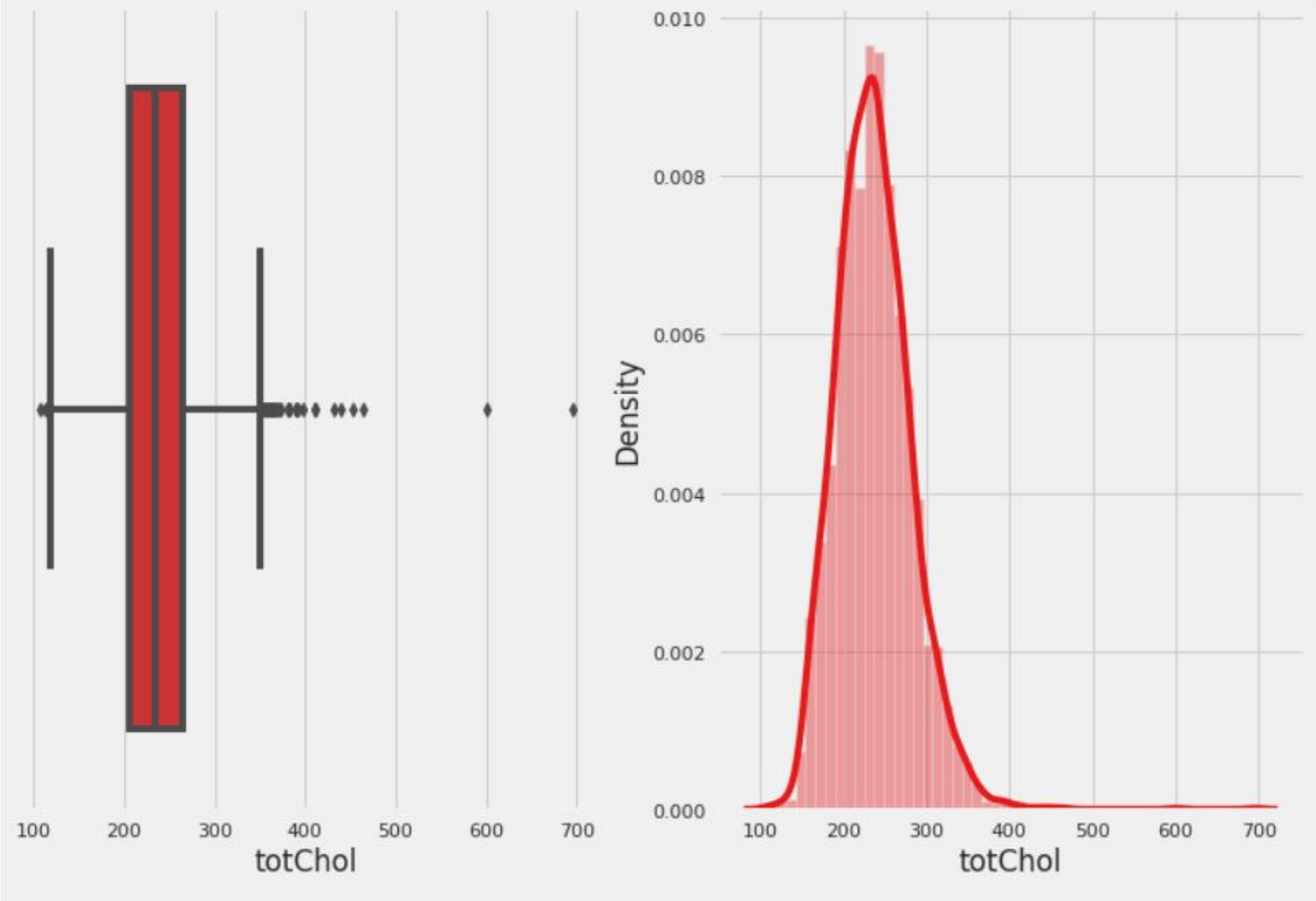
✓ ***Outliers:*** *After imputing the missing values. We detected the outliers in our dataset using boxplot. Later on we handled those in feature scaling section, where we used Robust scaler.*

✓ *After Imputing the missing values and detecting the outliers we are done with cleaning up part.*

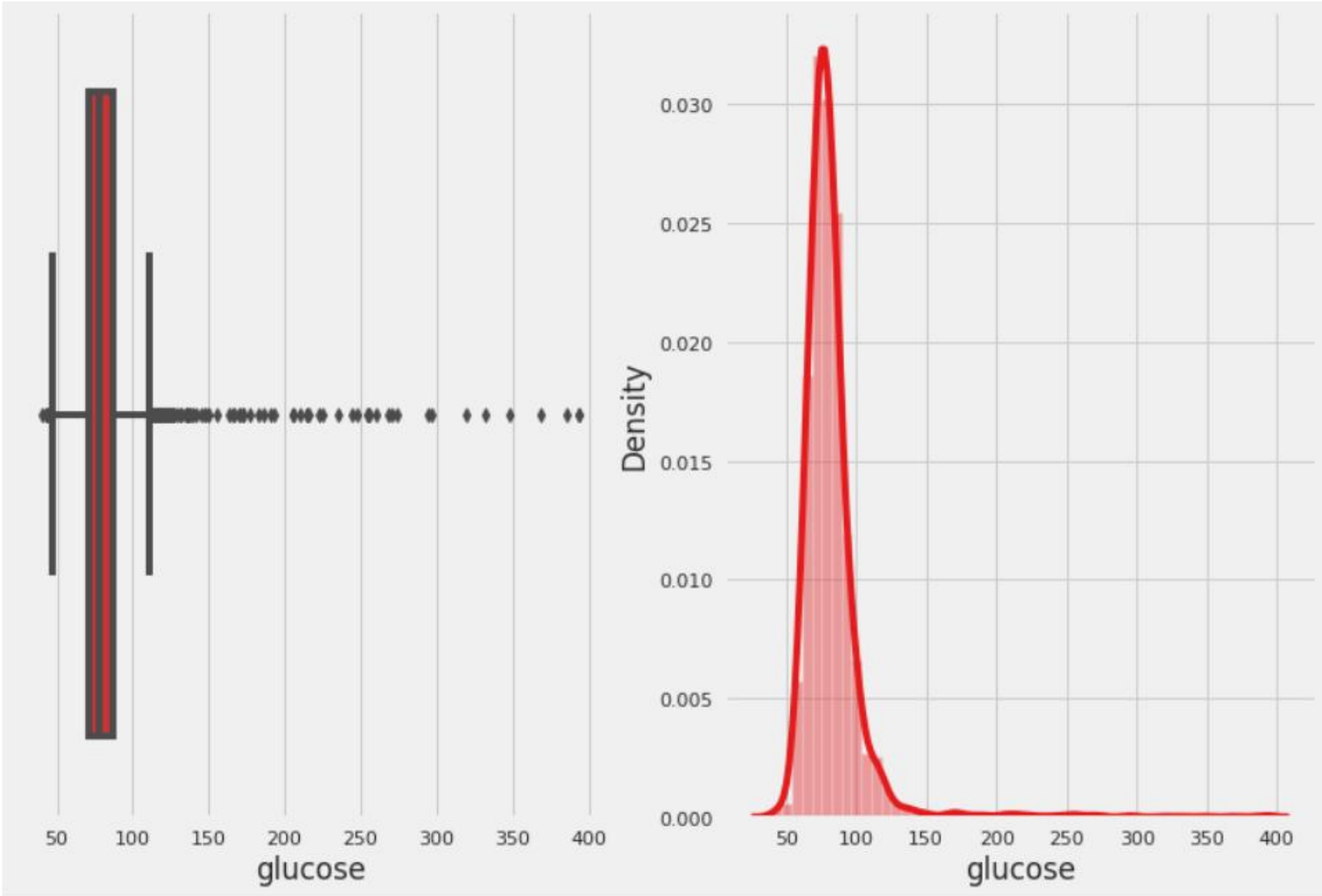
The box-plot and distribution plot of Competition distance shown.



The box-plot and distribution plot of Total Cholesterol shown below.



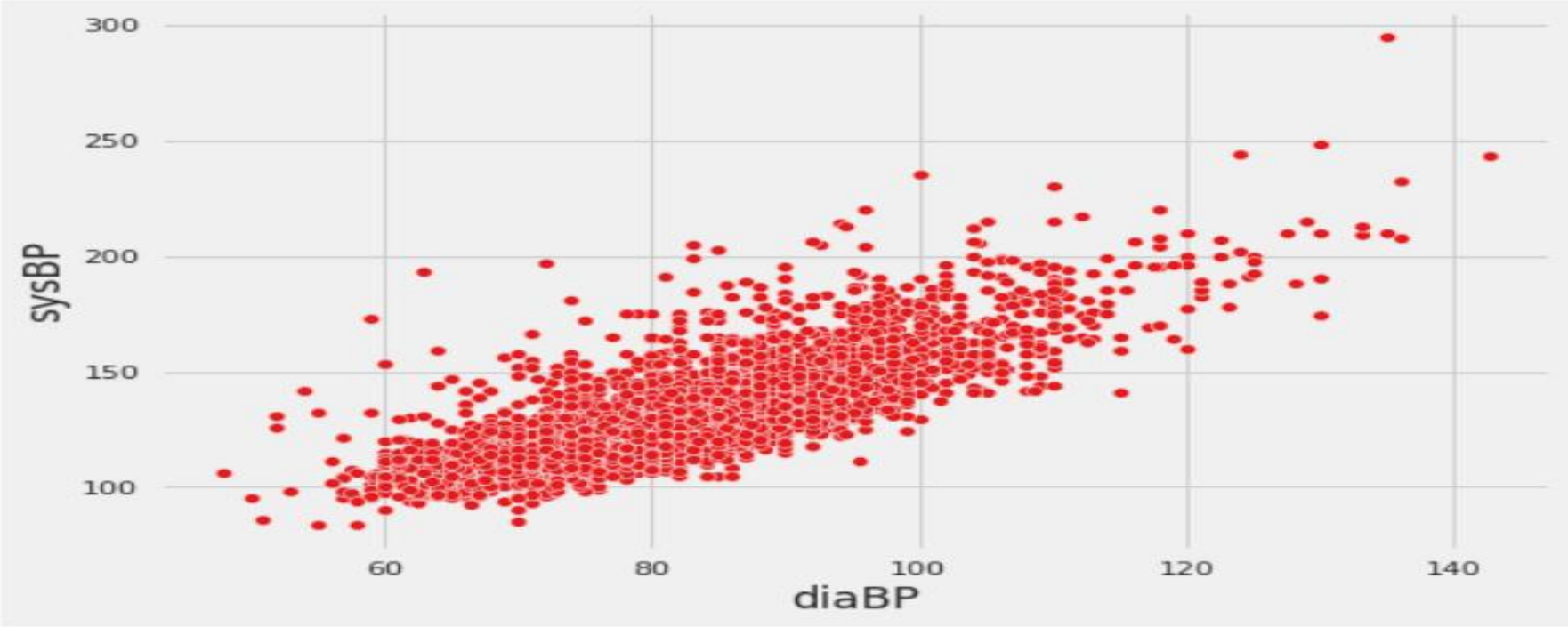
The box-plot and distribution plot of glucose shown below.



3. Feature Engineering:

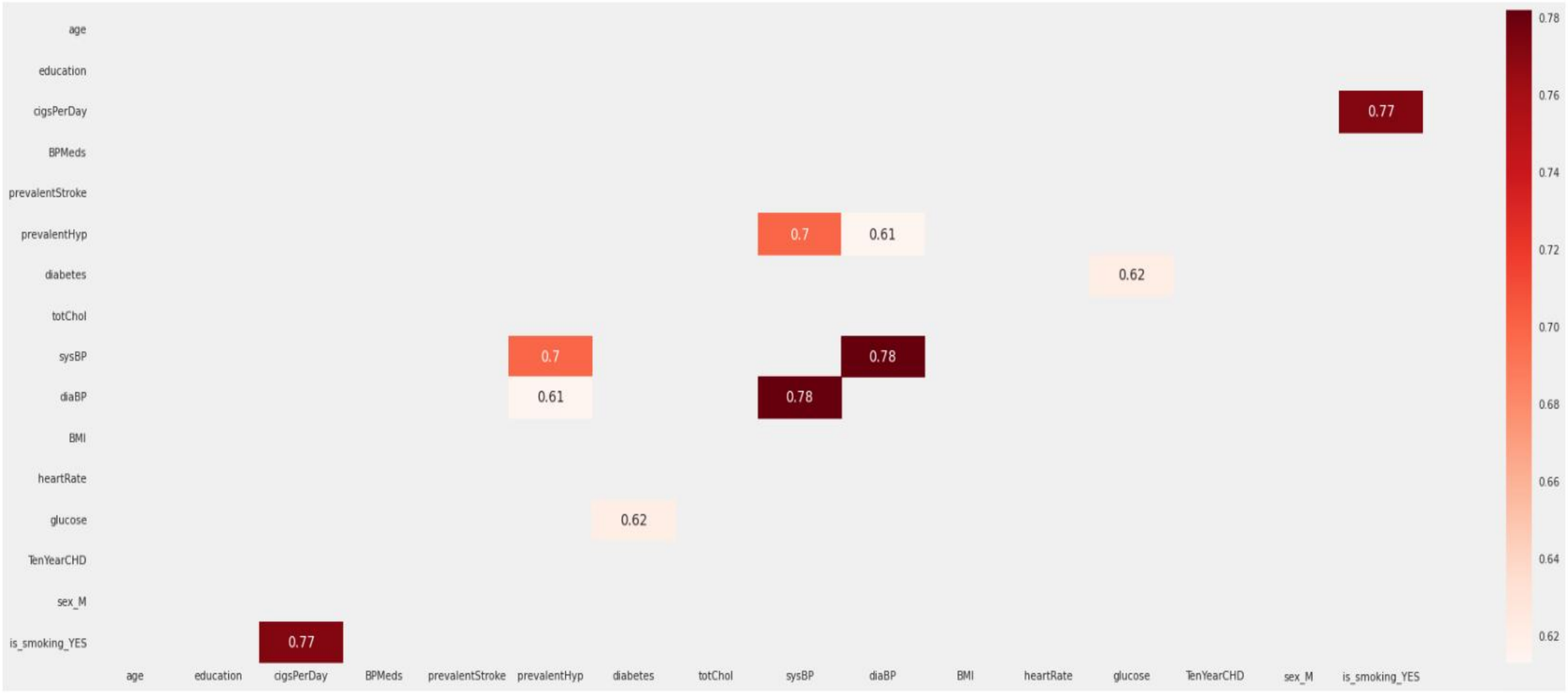
✓ ***Feature Encoding:*** *In feature engineering part, we encoded categorical feature. We applied ordinal encoding on ordinal categorical features and one-hot encoding on nominal categorical features.*

✓ ***Features Construction:*** *In feature construction, As we can see that from distribution plot and scatterplot that systolic blood pressure and diastolic blood pressure are highly correlated. The same we confirmed from heatmap and found the pearson correlation between these two variables to be 78%. So we decided to combine these two variables and formed new variable AvgBP.*



✓ ***Removing Multicollinearity:*** *We checked the multicollinearity among the independent variables using heatmap and we removed that features which were highly correlated.*

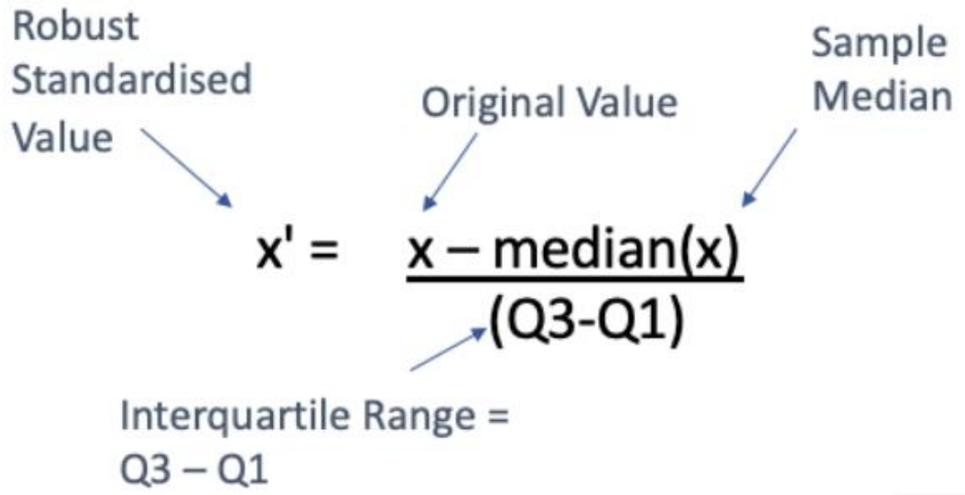
The heatmap of variables shown, having more than 50 % correlation coefficient among themselves.



4. Pre-Processing:

- ✓ ***Feature Scaling:*** *After removing multicollinearity, we scaled the features using standard-scaler, and robust-scaler(for continuous features). After Standardization, we shift the mean of all the independent variable to 0 and their Standard deviation to 1, thereby making the distribution standard normal. Using robust scaler, we are handling all the outliers present in our data.*

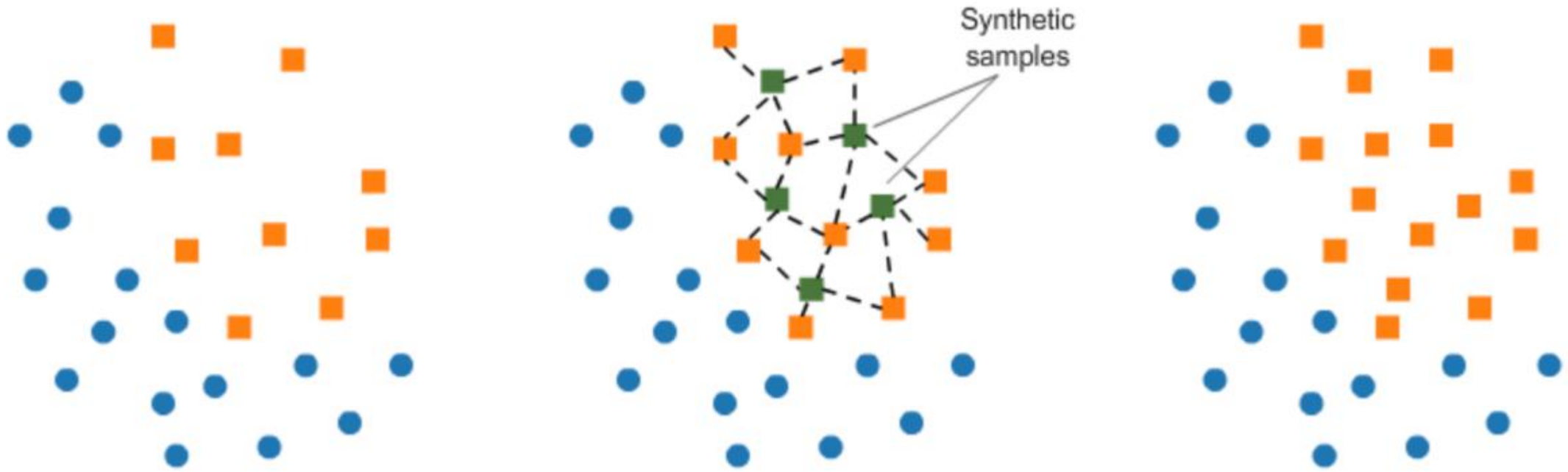
$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$



Synthetic Minority Oversampling Technique (SMOTE)

This technique generates synthetic data for the minority class.

SMOTE (Synthetic Minority Oversampling Technique) works by randomly picking a point from the minority class and computing the k-nearest neighbors for this point. The synthetic points are added between the chosen point and its neighbors.



The value counts of target variable before applying SMOTE.

```
1 y.value_counts()
```

```
0    2879
1     511
Name: TenYearCHD, dtype: int64
```

- *As we can see that our dataset has heavy imbalance in target column. We need to handle this using SMOTE.*

The value counts of target variable after applying SMOTE.

```
1 # counting the categories values.
2
3 y_sm.value_counts()
```

```
1    2879
0    2879
Name: TenYearCHD, dtype: int64
```

As we can see that after applying SMOTE, now the target column is balanced.

5. Train_Test_Split:

✓ ***Train Test Split:*** *After Pre-processing data, we split them into training and testing dataset in the ratio of 80:20 respectively using train_test_split method.*

```
[ ]  # Applying train_test_plit to split dataframe in train and test.
     # Testing size is 20% of the whole dataframe.

     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)
```

```
1 # CHecking the shape of X_train, X_test, y_train, and y_test dataset.
2
3 print("Shape of X_train : ",X_train.shape)
4 print("Shape of X_test : ",X_test.shape)
5 print("Shape of y_train : ",y_train.shape)
6 print("Shape of y_test : ",y_test.shape)
```

```
Shape of X_train :  (4606, 13)
Shape of X_test :  (1152, 13)
Shape of y_train :  (4606,)
Shape of y_test :  (1152,)
```

6. Model Implementation:

1. ***Logistic Regression:*** *Logistic regression analysis is used to predict the value of a target variable based on the value of independent variable(feature) using sigmoid curve by optimizimg the logit function.*

In **Logistic Regression**, the log-odds of a categorical response being "true" (1) is modeled as a linear combination of the features:

$$\log\left(\frac{p}{1-p}\right) = w_0 + w_1 x_1, \ldots, w_j x_j$$
$$= w^T x$$

where:

- w_0 is the intercept term, and w_1 to w_j represents the parameters for all the other features (a total of j features).
- By convention of we can assume that $x_0 = 1$, so that we can re-write the whole thing using the matrix notation $w^T x$.

This is called the **logit function**. The equation can be re-arranged into the **logistic function**:

$$p = \frac{e^{w^T x}}{1 + e^{w^T x}}$$

Or in the more commonly seen form:

$$h_w(x) = \frac{1}{1 + e^{-w^T x}}$$

✓ ***Score card of logistic regression:***

```
1 ## Classification report of Logistic Regression
2
3 print(classification_report(y_test, y_pred_log_reg))
```

	precision	recall	f1-score	support
0	0.65	0.68	0.67	553
1	0.69	0.67	0.68	599
accuracy			0.67	1152
macro avg	0.67	0.67	0.67	1152
weighted avg	0.68	0.67	0.67	1152

✓ ***Score card of KNN-Classifier:***

```
1 print(classification_report(y_test, y_pred_knn_grid))
```

	precision	recall	f1-score	support
0	0.96	0.69	0.80	553
1	0.77	0.97	0.86	599
accuracy			0.84	1152
macro avg	0.87	0.83	0.83	1152
weighted avg	0.86	0.84	0.83	1152

✓ ***Score card of support vector classifier:***

```
1 # displaying the classification report of SVC after doing cross validation.
2
3 print(classification_report(y_test, y_pred_svc_grid))
```

```
              precision    recall  f1-score   support

           0       0.77      0.77      0.77       553
           1       0.78      0.78      0.78       599

    accuracy                           0.78      1152
   macro avg       0.78      0.78      0.78      1152
weighted avg       0.78      0.78      0.78      1152
```

✓ ***Score card of Naïve Bayes classifier:***

```
1 # displaying the classification report of Naive Bayes Classifier,
2
3 print(classification_report(y_test, y_pred_naive_byes))
```

```
              precision    recall  f1-score   support

           0       0.54      0.91      0.68       553
           1       0.78      0.30      0.43       599

    accuracy                           0.59      1152
   macro avg       0.66      0.60      0.56      1152
weighted avg       0.67      0.59      0.55      1152
```

✓ ***Score card of Decision Tree Classifier:***

```
1 # Showing the classification report of Decision Tree Classification
2
3 print(classification_report(y_test, y_pred_dtree))
```

```
              precision    recall  f1-score   support

           0       0.84      0.80      0.82       553
           1       0.82      0.86      0.84       599

    accuracy                           0.83      1152
   macro avg       0.83      0.83      0.83      1152
weighted avg       0.83      0.83      0.83      1152
```

✓ ***Score card of Random Forest Classifier:***

```
# SHowing the classification report of randomforestclassifier.

print(classification_report(y_test, y_pred_random_frst))
```

	precision	recall	f1-score	support
0	0.86	0.92	0.89	553
1	0.92	0.86	0.89	599
accuracy			0.89	1152
macro avg	0.89	0.89	0.89	1152
weighted avg	0.89	0.89	0.89	1152

✓. *Score Card of XGBoost Classifier:*

```
2
3 print(classification_report(y_test, y_pred_xgbost))
```

	precision	recall	f1-score	support
0	0.80	0.91	0.85	553
1	0.90	0.79	0.84	599
accuracy			0.85	1152
macro avg	0.85	0.85	0.85	1152
weighted avg	0.85	0.85	0.85	1152

❑Model Explainability

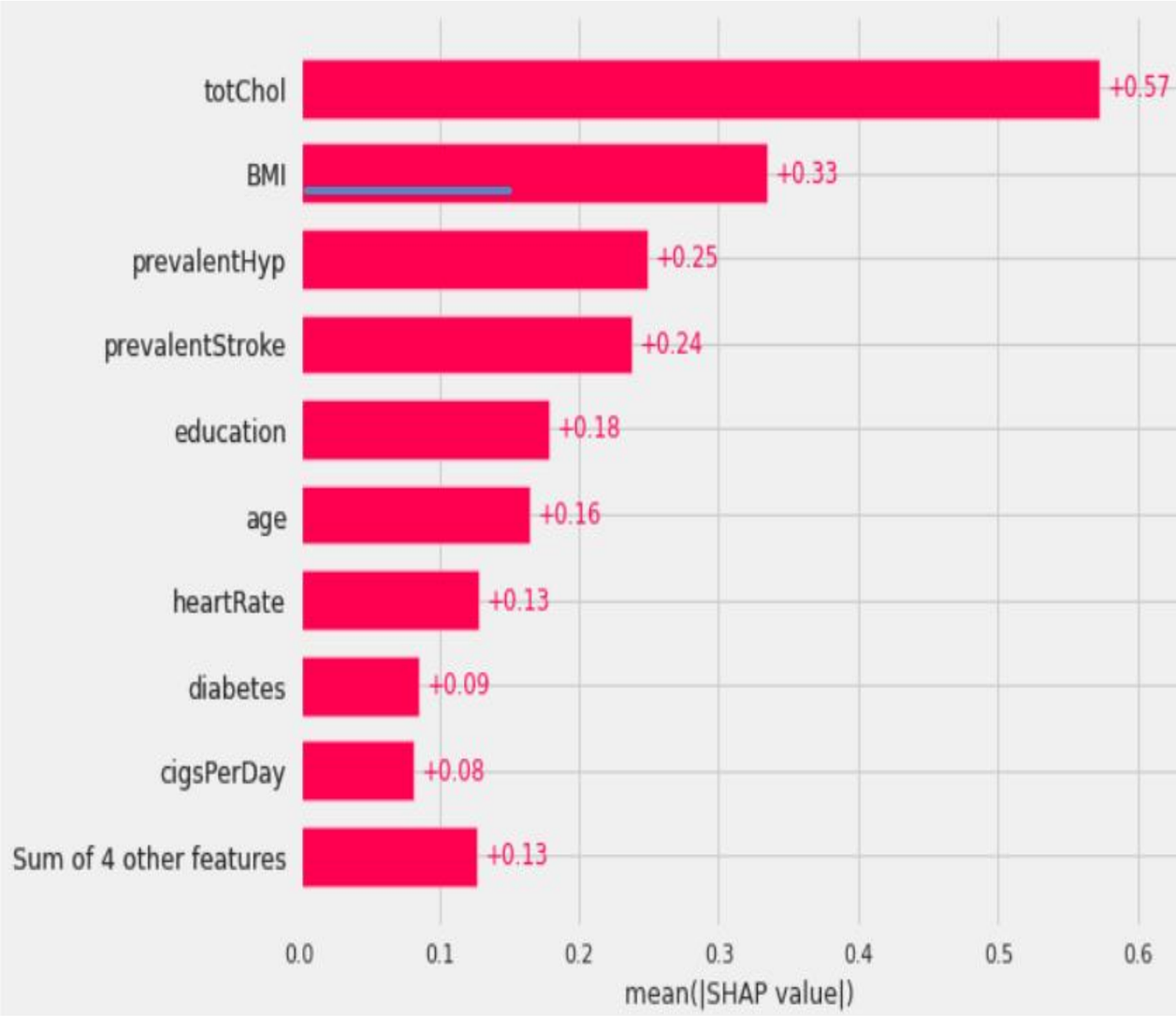
Model Explainability:

- Explainability in machine learning means that you can explain what happens in your model from input to output. It makes models transparent and solves the black box problem. Explainability is the degree to which a human can understand the cause of a decision or the degree to which a human can consistently predict ML model results. Explainability and interpretability are often used interchangeably. Although they have the same goal to understand the model.

Model Explainability Using SHAP (SHapley Additive exPlanations)

SHAP (SHapley Additive exPlanations) is a game theoretic approach to explain the o utput of any machine learning model. It connects optimal credit allocation with loca l explanations using the classic Shapley values from game theory and their related ex tensions (see papers for details and citations).

Model Explainability of XGBoost Classifier



✓As we can see that the Total Cholesterol is the most important variable having the highest weightage. It means that it is contributing the most in predicting the target.

✓We have noticed that cigarettes per day, and diabetes are one of the least contributer in predicting the target.

Challenges:

- *The dataset in the problem was heavily imbalanced with respect to the target variable. There are 2879 records classified as not prone to Coronary heart disease as opposed to 511 records having the risk of coronary heart disease.*

  We solved this issue by using SMOTE technique which creates synthetic records by using KNN. So our final dataset had equal proportions of both categories of target.

- *There were a lot of outliers in numeric columns, which could have badly affected the prediction. However we dealt them with robust scaler(transformation technique).*

- *The data was highly right skewed, on which logistic regresssion does not work well. So we standardized it(made the distribution standard normal) using standard-scaler.*

❑ Conclusions:

- *Total cholesterol was found to be the most important feature(using xgboost classifier), which is contributing the highest in predicting the target variable.*
- *We found 67% accuracy through logistic regression.*
- *The ensembles of decision tree i.e. Random forest gave us the highest accuracy i.e. 89% in predicting the target variable.*
- *However through Xgboost classifier, we got 85% accuracy.*
- *So we found Random forest to be the best performing algorithm.*

AI

THANK YOU